The Role of Business Processes in Service Oriented Architectures

W.M.P. van der Aalst
Eindhoven University of Technology,
Department of Mathematics and Computer Science,
P.O.Box 513, NL-5600 MB, Eindhoven, The Netherlands
E-mail: w.m.p.v.d.aalst@tue.nl

F. Leymann
Universität Stuttgart,
Institut für Architektur von Anwendungssystemen,
Universitätsstraße 38, 70569 Stuttgart, Germany
E-mail: frank.leymann@iaas.uni-stuttgart.de

W. Reisig
Humboldt-Universität zu Berlin,
Institut für Informatik,
Unter den Linden 6, 10099 Berlin, Germany
E-mail: reisig@informatik.hu-berlin.de

Abstract: Service oriented architectures, an emerging paradigm for designing and implementing business collaborations within and across organizational boundaries, are currently of interest to both software vendors and scientists. In this paradigm, the functionality provided by business applications is encapsulated within web services: software components described at a semantic level, which can be invoked by application programs or by other services through a stack of Internet standards including HTTP, XML, SOAP, WSDL, and UDDI. Once deployed, web services provided by various organizations can be interconnected in order to implement business collaborations, leading to composite web services. Ultimately, these composed services are there to support business processes. Therefore, the relationship between business process management, workflow technology, and service oriented architectures is highly relevant. This is illustrated by the interest in the BPEL standard. Therefore, this special issue of the International Journal of Business Process Integration and Management addresses the link between processes and services. This paper introduces the articles in this special issue and provides an overview of the domain.

Keywords: web services, service oriented architectures, workflow management, business process management.

Reference to this paper should be made as follows: W.M.P. van der Aalst, F. Leymann, and W. Reisig (2007) 'The Role of Business Processes in Service Oriented Architectures', International Journal of Business Process Integration and Management, Vol. XX, Nos. XX, pp.XX–XX.

Biographical notes: Prof.dr.ir. Wil van der Aalst is a full professor of Information Systems at the Technische Universiteit Eindhoven (TU/e) working on topics like workflow management, process mining, Petri nets, business process management, process modelling, and process analysis. Prof.dr. Frank Leymann is a full professor of Computer Science at the University of Stuttgart, Germany. His research interests include service oriented computing and middleware, workflow- and business process management, programming in the large, transaction processing, integration technology, and architecture patterns. Prof.dr. Wolfgang Reisig is a full professor of the Humboldt-Universität zu Berlin, working on the theory of programming. He has specialized on topics like Petri nets, concurrency theory, distributed algorithms, and abstract state machines.

1 INTRODUCTION

More and more, applications are no longer built from scratch but by integrating pieces of software that have been constructed independently from each other. As a consequence, the various pieces of an application must be loosely coupled. Service oriented architectures (SOA) provide a general prescription and guidelines of how to loosely couple pieces of application functionality. Web services are a concrete instantiation of a service oriented architecture. Clearly, *business processes are essential when aggregating loosely coupled functions into new application functionality.* For the combination of business process technology and Web service technology the terms *choreography* and *orchestration* have been coined. These technologies are expected to become the foundational layer for tomorrow's information systems and are influencing already many application areas like Enterprise Application Integration, Software Engineering, Systems Management, Data Provisioning, Business Process Intelligence, and Business-To-Business, just to name a few.

The focus on supporting processes started already in the seventies when people like Skip Ellis (Ellis, 1979), Anatol Holt (Holt, 1985), and Michael Zisman (Zisman, 1977) already worked on so-called office information systems, which were driven by explicit process models. Today workflow management systems are readily available (Aalst and Hee, 2004; Leymann and Roller, 1999; Muehlen, 2004) and workflow technology is hidden in many applications, e.g., ERP, CRM, and PDM systems. However, their application is still limited to specific industries such as banking and insurance. Since 2000 there has been a growing interest in web services. This resulted in a stack of Internet standards (HTTP, XML, SOAP, WSDL, and UDDI) which needed to be complemented by a process layer. Initially, several vendors proposed competing languages, e.g., IBM proposed WSFL (Web Services Flow Language) (Leymann, 2001) building on FlowMark/MQSeries and Microsoft proposed XLANG (Web Services for Business Process Design) (Thatte, 2001) building on Biztalk. BPEL (Andrews et al., 2003; Alves et al., 2007) emerged as a compromise between both languages.

The *Business Process Execution Language for Web Services* (BPEL4WS, or BPEL for short) has become the defacto standard for implementing processes based on web services (Andrews et al., 2003; Alves et al., 2007). Version 1.1 of BPEL was presented in 2003 (Andrews et al., 2003) and has been adopted by many vendors. Systems such as Oracle BPEL Process Manager, IBM WebSphere Application Server Enterprise, IBM WebSphere Studio Application Developer Integration Edition, and Microsoft BizTalk Server support BPEL, thus illustrating the practical relevance of this language. In April 2007, Version 2.0 (WS-BPEL 2.0) was approved as an OASIS Standard (Alves et al., 2007). Although intended as a language for connecting web services, its application is not limited to cross-organizational processes. This is illustrated by additions such as BPEL4People (Kloppmann et al., 2005) emphasizing tasks executed by people rather than web services. It is expected that in the near future a wide variety of process-aware information systems (Dumas et al., 2005) will be realized using BPEL.

In discussions, Petri nets (Reisig and Rozenberg, 1998) and Pi calculus (Milner, 1999) are often mentioned as two possible formal languages that could serve as a basis for languages such as BPEL. Some vendors claim that their systems are based on Petri nets or Pi calculus and other vendors suggest that they do not need a formal language to base their system on. After several years the debate on a standard language for process support is still ongoing and it seems unrealistic that consensus on a single language will be reached.

Despite the lack of consensus, it is clear that web services and processes support are already a reality today. Moreover, it is clear that a seamless integration between business process management and service oriented architectures is needed. This is the reason that a Dagstuhl seminar with the title *The Role of Business Processes in Service Oriented Architectures* was organized. This seminar took place in July 2006. The seminar was attended by more than 40 experts from both academia and industry. Unlike most of such seminars there was a high participation from industry (in particular from organizations developing SOA-related software, e.g., IBM, SAP, Microsoft, Google, etc.). This illustrates the practical relevance of the topic.

This seminar was highly successful and resulted in valuable proceedings (Leymann et al., 2006). After a selection process based on the contributions for this proceeding, some participants of this seminar were invited to submit extended versions of their papers. This special issue is the result of the reviewing process that followed.

In the remainder, we first discuss the role of models in the context of business processes and services. Then we provide a short introduction to the seven papers selected for this special issue.

2 THE ROLE OF MODELS

In this section we discuss the role of models for supporting the development and management of process-aware services. The reason is that this was one of the central topics of the Dagstuhl seminar, i.e., most of the discussions where ultimately related to this. Researchers tend to put emphasis on the role of models while practitioners tend to focus more on industry standards and the actual implementation of concepts.

Despite the scepticism of some practitioners, models already play an important role in information systems today. Moreover, it is expected that the importance of models will increase. Models can be used to specify systems and processes and can be used to analyze important properties. In fact, some of today's information systems are already driven by models (cf. workflow management sys-

tems). Although the general vision of a "Model Driven Architecture" (MDA) is appealing, it is not yet feasible for many applications. Only in specific niches such as workflow technology, MDA is already a reality and has proven to be valuable. In the context of Enterprise Resource Planning (ERP) systems, i.e., the world of SAP, PeopleSoft, Oracle, etc., models play a less prominent role. These systems offer a workflow component, but most of their functionality is still hard-coded. The well-known reference model of SAP (Curran and Keller, 1997) contains 604 Event-driven Process Chains (EPCs), modelling all the different business processes supported by the R/3 system. However, these EPC models are not used for enactment and serve merely as background information. It seems vital that ERP systems like SAP commence using models as a starting point, rather than just as a means to document things afterwards. It seems particularly interesting to use *configurable* process models as a starting point given the need for customization (Rosemann and Aalst, 2007).

Models are highly relevant for the *enactment* of processes. The core idea of classical workflow management systems is to automate processes based on process models (Aalst and Hee, 2004; Aalst, 2004; Leymann and Roller, 1999; Georgakopoulos et al., 1995). Also languages like BPEL allow for the enactment of processes. However, models can also be used *to analyze* processes. Using Fig. 1 we would like to zoom in on the types of analysis already possible today. We will focus on two types of analysis: (1) analysis at *design-time* and (2) analysis at *runtime*. At design-time, the only basis for analysis is a model, e.g., a workflow (re)design. At runtime, one can also observe the actual behaviour and use this as input for analysis.

Figure 1 shows an overview of the different types of analysis. To explain the diagram let us first consider the top part showing the interaction between the "world" and some (software) system. Any information system, and in particular a service oriented architecture, ultimately interacts with some physical environment; otherwise it serves no purpose. The system may support or control all kinds of processes taking place in the real world. Moreover, most systems also record events taking place inside and outside the system as indicated by the arrow connecting the "world" to event logs via the (software) system. Today's information systems log enormous amounts of events. Classical workflow management systems (e.g., Staffware), ERP systems (e.g., SAP), case handling systems (e.g., FLOWer), PDM systems (e.g., Windchill), CRM systems (e.g., Microsoft Dynamics CRM), middleware, hospital information systems (e.g., Chipsoft), etc. provide very detailed information about the activities that have been executed. Even embedded systems are connected to the Internet today, thus allowing for unprecedented streams of data. On the other hand, models play a prominent role as indicated in Fig. 1. Examples of models are process models such as BPMN diagrams, EPCs, Petri nets, BPEL specifications, UML activity diagrams, but also other types of models such as social networks, organizational charts, data models, etc. These models can be used to model the "world". However, they can also be used to model the system. In this context it is important to note that most information systems have a model of reality; i.e., a software system that has no "mental image" of the organizational context and the processes it should support is of limited use. It is often remarkable to see the resemblance between simulation models and workflow models. This supports the observation that information systems need to have a model of reality.

In an MDA or workflow setting, models are used to configure the information system as shown in Fig. 1. This approach is also used in the context of services, and the broad adoption of BPEL is a nice illustration of this. The analysis of processes in the context of services is possible but the practical application is still limited. Therefore, we briefly discuss the six types of analysis indicated in Fig. 1.

The correctness, effectiveness, and efficiency of the business processes supported by a service oriented architecture are vital to the organization. A process definition which contains errors may lead to annoyed customers, back-log, damage claims, and loss of goodwill. Flaws in the design of a process definition may also lead to high throughput times, low service levels, and a need for excess capacity. This is why it is important to analyze a process before it is put into production. As shown in Fig. 1, there are three types of design-time analysis:

- *validation*, i.e., testing whether the process behaves as expected,
- *verification*, i.e., establishing the correctness of a process definition, and
- *performance analysis*, i.e., evaluating the ability to meet requirements with respect to throughput times, service levels, and resource utilization.

Validation can be achieved by interactive simulation: a number of fictitious cases are fed to the system to see whether they are handled well. For verification and performance analysis more advanced analysis techniques are needed. Fortunately, many powerful analysis techniques have been developed and some of the corresponding tools have become mature in recent years. As an example, consider the Petri-net-based techniques and tools available for the modeling and analysis of workflows (Aalst and Hee, 2004; Aalst, 2004; Lohmann et al., 2006; Massuthe et al., 2005).

For the analysis of business processes supported by a service oriented architecture, it is also possible to use the event logs present as shown in Fig. 1. There are two possible ways of obtaining event logs: (1) monitoring messages exchanged between services (e.g., tapping of SOAP messages) and (2) having logging facilities at the middleware layer. A nice example of the second category is IBM's Common Event Infrastructure (CEI) that logs events in a systematic manner. *Process mining* (Aalst et al., 2007) has emerged as a way to analyze services and their actual use based on the event logs they produce. Assuming that we

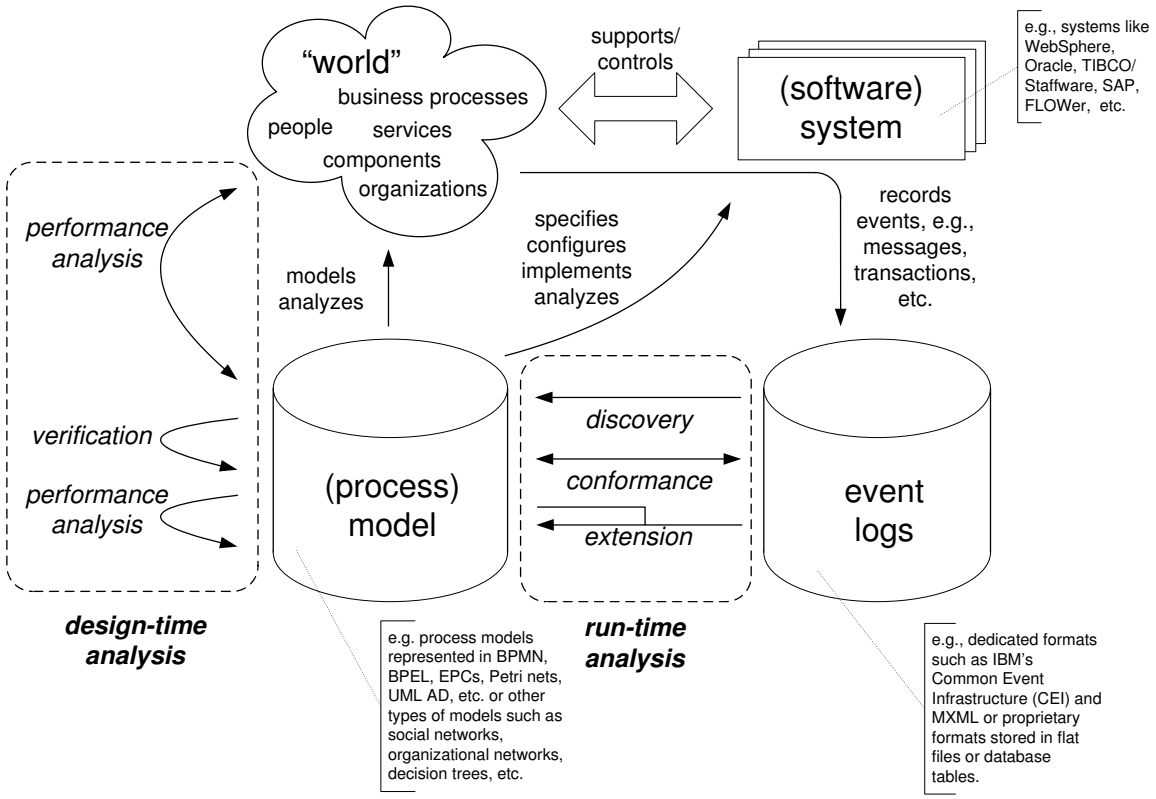

Figure 1: The relationships between reality, systems, logs, and models and the different types of design-time and run-time analysis.

are able to log events in a service oriented architecture, a wide range of *process mining techniques* comes into reach. The basic idea of process mining is to learn from observed executions of a process. As shown in Fig. 1, three types of process mining (i.e., runtime analysis) can be identified:

- *discovery*, i.e., automatically extracting models from event logs (e.g., constructing a Petri net that is able to reproduce the observed behaviour),
- *conformance*, i.e., checking whether the modelled behaviour matches the observed behaviour (e.g., Does the real service behaviour conform to some abstract BPEL specification?), and
- *extension*, i.e., extending an existing model by projecting information extracted from the logs onto some initial model (e.g., show bottlenecks in a process model by analyzing the event log).

All three types of process mining are meaningful in a service oriented architecture.

Based on Fig. 1 we discussed two types of analysis: analysis at design-time and analysis at runtime. An elaborate discussion of the different types of analysis is outside the scope of this paper. However, we would like to conclude with the following two statements:

- *Verification of real-life processes in service oriented architectures has become a reality!* It is possible to verify large sets of complicated models and these efforts pay off because often many errors are found. For example, the 604 EPCs of the SAP reference models can be easily analyzed and many design errors are uncovered by doing so (Mendling et al., 2006a,b). It is also possible to verify real-life processes specified in terms of BPEL (cf. tools such as WofBPEL, BPEL2oWFN, and Fiona).

- *The abundance of event logs in service oriented architectures allows for new and exciting forms of process analysis.* Process mining techniques can use this information in various ways. It may be used to discover the way that people and services really work, it may be used to find out where there are deviations in a distributed environment, it may be used to support people in performing their duties, and ultimately it can be used for all kinds of process improvement. Tools such as ProM support all these types of analysis.

In this section, we discussed the role of models in the context of business process management and service oriented architectures. Most of the discussions during the Dagstuhl seminar were related to this topic. For the effective utilization of the results of scientific work in industry, shared models of the utmost importance.

3 SELECTED PAPERS

As indicated in the introduction, this special issue contains seven extended versions of papers presented at/resulting from the Dagstuhl seminar with the title *The Role of Business Processes in Service Oriented Architectures* (Leymann et al., 2006). The selected papers demonstrate recent advances in business process management and web services, and cover aspects ranging from verification to pricing.

Peter Massuthe of the Humboldt-Universität zu Berlin and *Karsten Wolf* of the Universität Rostock describe an algorithm for matching services with *operating guidelines*. Their starting point is a service-oriented architecture where each party typically fulfils one of three possible roles: as a service provider, a service requester, and a service broker. This comes with three operations: publish, find, and bind. . The authors focus on the first two operations. In the paper, they use nondeterministic automata to model services and their interaction, and suggest operating guidelines as a convenient and intuitive artefact to be *published*. In their approach, the *find* operation reduces to a matching problem between the requester's service and the published operating guidelines. The paper provides solid theoretical results and shows that the distributed nature of services poses interesting scientific problems which are relevant from a practical point of view.

Wil van der Aalst, *Michael Beisiegel*, *Kees van Hee*, *Dieter König*, and *Christian Stahl* present a SOA-based architecture framework. On the one hand, the paper takes a conceptual approach by identifying the key concepts in service-oriented architectures. On the other hand, the authors provide explicit links to contemporary industry standards, especially to the Service Component Architecture (SCA) which is actively supported by IBM. The interesting fusion between a more academic conceptualization and a more industry-oriented view on the world is explained by the mixture of authors (partly from IBM and partly from academia). The paper provides a meta model to identify and structure the main elements of a SOA. Moreover, a graphical notation is proposed. Through the resulting framework concepts such as *wiring, correlation, and instantiation* can be clarified. The overall goal of the paper is to demystify some of the SOA terms and concepts.

Dominik Kuropka and *Mathias Weske* of the Hasso Plattner Institut of IT Systems Engineering in Potsdam argue in their contribution that the full potential of service-based software architectures can only be achieved by extending the currently available syntactic descriptions of services with semantic descriptions. The authors start by providing a brief introduction to the state of the art in service oriented architectures. Then they show that semantic descriptions are needed for matchmaking and binding of services as well as integration and composition of services. Based on this approach, a semantic service platform is introduced that implements dynamic matchmaking, composition and binding of semantically described services. Moreover, an application scenario is given.

Uwe Zdun and *Schahram Dustdar* of the Information Systems Institute of the Technical University of Vienna propose to integrate process-driven SOA models via a model-driven software development approach that is based on proven practices documented as software patterns. In their paper, the authors focus on model integration by introducing an approach that is based on a common meta-meta-model from which concrete meta-models for Domain Specific Languages (DSLs) are derived. In the different DSLs and their respective meta-models, proven practices (described as software patterns) are specified as modelling primitives, and their constraints can be validated for all instances of all different meta-models. Examples are given, showing how to integrate message flow models, business process models, and architectural models.

Björn Axenath, *Ekkart Kindler*, and *Vladimir Rubin* of the University of Paderborn present a paper on AMFIBIA. AMFIBIA is a meta-model that formalizes the essential aspects and concepts of business processes in a formalism-independent manner. Its core is formed by concepts such as *business process model, case, task* and *activity*. A business process model consists of a set of tasks. A case is an instance of a particular business process model. While a case is executed, different tasks will be instantiated to activities; each activity corresponds to exactly one task. AMFIBIA is not restricted to a fixed set of aspects and attempts to capture the interaction among the different aspects and concepts. As a proof-of-concept the authors implemented a prototype of a formalism-independent workflow engine based on AMFIBIA.

Oliver Günther and *Gerrit Tamm* of the Humboldt-Universität zu Berlin and *Frank Leymann* of Universität Stuttgart focus on the pricing of web services in their contribution. They have studied pricing mechanisms for composite web services and their integration into new or existing web service standards. In the paper, the results of an online experiment are presented. In this experiment 242 test persons were confronted with a variety of choices and decisions relating to web service markets and service composition. One of the insights is that users are not willing to pay for aggregation by a third party. This shows that different pricing models need to be taken into account. Moreover, existing intermediaries like UDDI need to be extended to support market transactions, including more complex and more flexible ones, such as negotiations or auctions.

Guadalupe Ortiz and *Juan Hernández* of the University of Extremadura propose a model-driven UML-based approach for web services and their extra-functional properties. The goal is to add extra-functional properties to modelled services in order to support the entire web service life cycle. The authors propose a Model-Driven Architecture (MDA) approach where services and properties are initially described at the PIM (Platform Independent Model) level using UML. Then the initial PIM is converted into four specific models, the first one models JAX-RPC-based services and the other three model extra-functional properties. Finally, the authors advocate the use of aspect-

oriented techniques to provide a better decoupling.

It should be noted that most of the papers in this special issue focus on the modelling of services and processes. This illustrates the central role of models as already discussed in Fig. 1.

ACKNOWLEDGMENT

We would like to thank Satish Thatte for co-organizing the 2006 Dagstuhl seminar on "The Role of Business Processes in Service Oriented Architectures".

REFERENCES

Aalst, W. van der (2004). Business Process Management Demystified: A Tutorial on Models, Systems and Standards for Workflow Management. In Desel, J., Reisig, W., and Rozenberg, G., editors, *Lectures on Concurrency and Petri Nets*, volume 3098 of *Lecture Notes in Computer Science*, pages 1–65. Springer-Verlag, Berlin.

Aalst, W. van der and Hee, K. (2004). *Workflow Management: Models, Methods, and Systems*. MIT press, Cambridge, MA.

Aalst, W. van der, Reijers, H., Weijters, A., van Dongen, B., Medeiros, A., Song, M., and Verbeek, H. (2007). Business Process Mining: An Industrial Application. *Information Systems*, 32(5):713–732.

Alves, A., Arkin, A., Askary, S., Barreto, C., Bloch, B., Curbera, F., Ford, M., Goland, Y., Guzar, A., Kartha, N., Liu, C., Khalaf, R., Koenig, D., Marin, M., Mehta, V., Thatte, S., Rijn, D., Yendluri, P., and Yiu, A. (2007). Web Services Business Process Execution Language Version 2.0 (OASIS Standard). WS-BPEL TC OASIS, http://docs.oasis-open.org/wsbpel/2.0/wsbpel-v2.0.html.

Andrews, T., Curbera, F., Dholakia, H., Goland, Y., Klein, J., Leymann, F., Liu, K., Roller, D., Smith, D., Thatte, S., Trickovic, I., and Weerawarana, S. (2003). Business Process Execution Language for Web Services, Version 1.1. Standards proposal by BEA Systems, International Business Machines Corporation, and Microsoft Corporation.

Curran, T. and Keller, G. (1997). *SAP R/3 Business Blueprint: Understanding the Business Process Reference Model*. Upper Saddle River.

Dumas, M., Aalst, W. van der, and Hofstede, A. (2005). *Process-Aware Information Systems: Bridging People and Software through Process Technology*. Wiley & Sons.

Ellis, C. (1979). Information Control Nets: A Mathematical Model of Office Information Flow. In *Proceedings of the Conference on Simulation, Measurement and Modeling of Computer Systems*, pages 225–240, Boulder, Colorado. ACM Press.

Georgakopoulos, D., Hornick, M., and Sheth, A. (1995). An Overview of Workflow Management: From Process Modeling to Workflow Automation Infrastructure. *Distributed and Parallel Databases*, 3:119–153.

Holt, A. W. (1985). Coordination Technology and Petri Nets. In Rozenberg, G., editor, *Advances in Petri Nets 1985*, volume 222 of *Lecture Notes in Computer Science*, pages 278–296. Springer-Verlag, Berlin.

Kloppmann, M., Koenig, D., Leymann, F., Pfau, G., Rickayzen, A., Riegen, C., Schmidt, P., and Trickovic, I. (2005). WS-BPEL Extension for People - BPEL4People. IBM Corporation, http://www-128.ibm.com/developerworks/webservices/library/specification/ws-bpel4people/.

Leymann, F. (2001). Web Services Flow Language, Version 1.0.

Leymann, F., Reisig, W., Thatte, S., and van der Aalst, W. van der, editors (2006). *The Role of Business Processes in Service Oriented Architectures*, number 6291 in Dagstuhl Seminar Proceedings. Internationales Begegnungs- und Forschungszentrum fuer Informatik (IBFI), Schloss Dagstuhl, Germany.

Leymann, F. and Roller, D. (1999). *Production Workflow: Concepts and Techniques*. Prentice-Hall PTR, Upper Saddle River, New Jersey, USA.

Lohmann, N., Massuthe, P., Stahl, C., and Weinberg, D. (2006). Analyzing Interacting BPEL Processes. In Dustdar, S., Faideiro, J., and Sheth, A., editors, *International Conference on Business Process Management (BPM 2006)*, volume 4102 of *Lecture Notes in Computer Science*, pages 17–32. Springer-Verlag, Berlin.

Massuthe, P., Reisig, W., and Schmidt, K. (2005). An Operating Guideline Approach to the SOA. *Annals of Mathematics, Computing & Teleinformatics*, 1(3):35–43.

Mendling, J., Aalst, W. van der, Dongen, B. van, and Verbeek, E. (2006a). Errors in the SAP Reference Model. *BPTrends*, 4(6):1–5.

Mendling, J., Moser, M., Neumann, G., Verbeek, H., Dongen, B. van, and Aalst, W. van der (2006b). Faulty EPCs in the SAP Reference Model. In Dustdar, S., Faideiro, J., and Sheth, A., editors, *International Conference on Business Process Management (BPM 2006)*, volume 4102 of *Lecture Notes in Computer Science*, pages 451–457. Springer-Verlag, Berlin.

Milner, R. (1999). *Communicating and Mobile Systems: The Pi-Calculus*. Cambridge University Press, Cambridge, UK.

Muehlen, M. zur (2004). *Workflow-based Process Controlling: Foundation, Design and Application of workflow-driven Process Information Systems*. Logos, Berlin.

Reisig, W. and Rozenberg, G., editors (1998). *Lectures on Petri Nets I: Basic Models*, volume 1491 of *Lecture Notes in Computer Science*. Springer-Verlag, Berlin.

Rosemann, M. and Aalst, W. van der (2007). A Configurable Reference Modelling Language. *Information Systems*, 32(1):1–23.

Thatte, S. (2001). XLANG Web Services for Business Process Design.

Zisman, M. (1977). *Representation, Specification and Automation of Office Procedures*. PhD thesis, University of Pennsylvania, Warton School of Business.